REMARKS

The application includes claims 1-4 and 6-20 prior to entering this amendment.

The examiner objects to claims 17-20 because of informalities.

The examiner rejects claims 20 under 35 U.S.C. § 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

The examiner rejects claims 1-4 and 6-20 under 35 U.S.C. § 102(e) as being anticipated by Myszne (U.S. Patent 7,106,859 B2).

The applicants amend claims 1, 8, 13, 17, and 20 and cancels claims 3 and 15 without prejudice. The applicants had previously canceled claim 5. The application remains with claims 1-2, 4, 6-14, and 16-20 after entering this amendment.

The applicants add no new matter and request reconsideration.

Claim Objections

The examiner objects to claims 17-20 because claim 17 recites "a predict logic block ... responsive an immediately preceding state of the subset" instead of —a predict logic block... responsive to an immediately preceding state of the subset—. The applicants amend claim 17 to obviate the examiner's objection.

Claim Rejections Under § 112

The examiner rejects claim 20 as indefinite because it is unclear to which version of the IEEE 802.11a standard it recites. The applicants amend claim 20 to obviate the examiner's rejection.

Claim Rejections § 102

The examiner rejects claims 1-4, 6-13 and 17-20 as old over Myszne. The applicants disagree for the reasons that follow.

The present application describes a parallel scrambler/descrambler system for use in wireless communication devices. The parallel scrambler/descrambler system scrambles sets of data bits using subsets of recurring sequence of scrambler bits. A self-synchronous scrambler generates repeating sequences of scrambler bits regardless of the initial stage of the scrambler. The scrambler predicts the next n states based on the current state. A parallel logic operation between preselected bits of the current state will yield the required values used in scrambling an incoming parallel data set. A parallel logic operation between the required values and the incoming data set results in the scrambled output data.

Myszne describes a parallel data scrambler in which a scrambler value table 22 stores the scrambler output values produced by the generator 10 (i.e., the entire m-sequence) multiple times and selector 24 may select a set of N values which it provides to parallel XOR unit 26 for use in scrambling N bits of the incoming data.[1] The scrambler value table 22 stores the 127 bit m-sequence plus one bit in multiple swaths.[2] "The table may be formed into at least two overlapping swaths of N columns, wherein each swath may store the m-sequence and the m-sequence of one swath is shifted from the m-sequence of a second swath....When the swath is finished, the scrambler may shift to another swath."[3]

Claim 1 recites *generating a next state of the subset by logically manipulating at least one bit of the current state of the subset with at least another bit of the current state of the subset.* Claim 8 recites *the subset being determined by logically manipulating at least one bit of a preceding state of the subset with at least another bit of the preceding state of the subset.* Claims 13 and 17 include similar language. "Once the 7 output bits have been generated, the scrambling bit sequence has to be incremented to the next state. Effectively, this step advances the bit sequence by 7 iterations if one were to think of the bit sequence is such that the next state of each element in the bit sequence can be expressed in terms of the current state of the bit sequence elements."[4] For example, the system generates the next state of the scramble bit sequence Nxt[0] by logically XORing x1 with x2 and x5, where x represents the current state of the bit sequence. In contrast, Myszne's scrambler shifts to another swath to get the next set of scrambler bits. No logical manipulation of at least one bit of a preceding state of the subset with at least another bit

---

[1] Myszne, column 2, lines 10-16.
[2] Myszne, column 2, lines 33-35.
[3] Myszne, abstract.

of the preceding state of the subset occurs when Myszne's system shifts from one swath to another. Note too that Myszne's generator 10 serially fills the value table 22 (or 50) for selection by the selector 24. The selector 24 merely selects N values "which it provides to parallel XOR unit 26 for use in scrambling N bits of the incoming data."[5]

Claim 1 recites *where a number of bits in the subset corresponds to a periodicity of the serial sequence of scramble bits.* Claims 8, 13, and 17 include similar language. The examiner indicates that Myszne discloses the element when it discloses that its generator 10 is "cyclic, reading the last row of swath 52 provides the first three values of the m-sequence, the next swath."[6] The applicants acknowledge that Myszne's generator 10 is cyclic, but disagree that the inclusion of additional bits at the end of the m-sequence (1 additional bit in table 22 and 3 additional bits in table 50) discloses the recited *a number of bits in the subset corresponds to a periodicity of the serial sequence of scramble bits.*
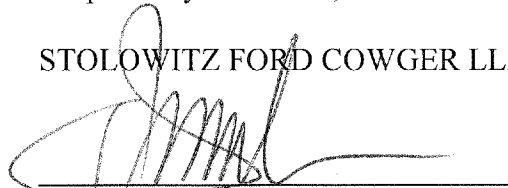
Conclusion

In view of the foregoing, applicants respectfully submit that claims 1-2, 4, 6-14, and 16-20 are allowable and ask the examiner to pass this application to allowance. If the examiner has any questions or believes that a telephone conference would expedite prosecution of this application, applicants encourage the examiner to call the undersigned at (503) 224-2170.

Customer No. 73552

Respectfully submitted,

STOLOWITZ FORD COWGER LLP

Graciela G. Cowger
Reg. No. 42,444

STOLOWITZ FORD COWGER LLP
621 SW Morrison Street, Suite 600
Portland, OR 97205
(503) 224-2170

---

[4] Amer, paragraph [0051].
[5] Myszne, column 2, lines 11-12.
[6] Office action dated 8/15/2007, page 4.